



(2)

Estimation of EOF Expansion Coefficients from Incomplete Data

DTIC
S ELECTED SEP 17 1992 D
C

J. D. Boyd
Oceanography Division
Ocean Science Directorate

E. P. Kennelly
Neptune Sciences, Inc.
Slidell, LA 70458

P. Pistek
Oceanography Division
Ocean Science Directorate



Approved for public release; distribution is unlimited. Naval
Oceanographic and Atmospheric Research Laboratory, Stennis Space
Center, Mississippi 39529-5004.

92 9 15 858

421485

92-25280



46p8

**These working papers were prepared for the timely
dissemination of information; this document does
not represent the official position of NOARL.**

Abstract

A least-squares minimization technique for estimating principal components (empirical orthogonal function expansion coefficients) from incomplete ocean profile data is developed, and its use for interpolating or extrapolating these profiles is described.

DTIC QUALITY INSPECTED 3

Accession For	
NTSC	<input checked="" type="checkbox"/>
VHS TAB	<input type="checkbox"/>
Unrestricted	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Approved/or	
Dist	Special
R-1	

Acknowledgments

This work was funded by the Office of Naval Technology Code 230 under Program Elements 0602435N and 0602314N, Dr. R. Doolittle, ONT Program Manager, and Dr. E. Franchi, Naval Research Laboratory Program Manager.

Contents

1. Introduction	1
2. Theory	2
3. Example	3
4. Coefficient Estimation Software	5
5. Concluding Remarks	5
6. References	7
Appendix A: FORTRAN Subprogram	21

Estimation of EOF Expansion Coefficients from Incomplete Data

1. Introduction

Among the earliest published reports on the use of empirical orthogonal functions (EOFs) in oceanography are included such articles as Moore (1974), Kundu, et al. (1975) and Davis (1976). Since then, the use of EOFs has become increasingly common for such purposes as describing and quantifying oceanic variability (e.g., Halliwell and Mooers, 1979), in examining ocean dynamics (e.g., Kundu, et al., 1975), in providing compact characterization of oceanographic data (e.g., Tolstoy, et al., 1991), and in examining interrelationships between surface and subsurface parameters (e.g., Carnes, et al., 1990). In this note we present a technique for using EOFs for interpolating or extrapolating missing data in ocean profiles. This technique should prove particularly useful for extrapolating shallow ocean data profiles -- say, from shallow CTD (conductivity-temperature-depth) casts or from expendable probes -- to deeper depths.

The theory behind EOF computations is straightforward. (See, for example, Lagerloef and Bernstein, 1988, for a particularly compact and lucid description.) As an aid to understanding our technique described in section 2, we present a summary of the EOF computation procedures here.

Suppose we have N profiles with measurements at K discrete depths. Each profile may be considered a vector f_n , with components f_{kn} . A set of perturbation profiles F_n is formed by subtracting the mean profile \bar{f} from each of the observed profiles. From these profiles the $K \times N$ matrix $[F]$ is created, having columns which are the N perturbation profiles f_n . Next the $K \times K$ covariance matrix $[R]$ is calculated from the $[F]$ matrix according to

$$[R] = (1/N) [F] [F]^T,$$

where $[F]^T$ is the transpose of $[F]$.

Then the eigenvalues λ_k and eigenvectors e_k are found for the matrix equation

$$[R] [E] = [L] [E].$$

$[L]$ is the matrix of eigenvalues

$$[L] = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_K \end{bmatrix}$$

and $[E]$ is the matrix of eigenvectors

$$[E] = [e_1 \ e_2 \ \dots \ e_m \ \dots \ e_K]$$

where e_m is a column vector representing the m th eigenmode. A maximum of K eigenmodes may be calculated.

The original data may be expanded in terms of these new basis functions e_m according to the relationship

$$[F] = [E] [C] \quad (1)$$

where the c_{mn} elements of $[C]$ are known as the expansion coefficients, the modal amplitudes, or the principal components. The coefficient matrix $[C]$ is computed from

$$[C] = [E]^T [F]. \quad (2)$$

The trace of the matrix $[R]$ is the total variance of the data set. Since $\text{Tr} [R] = \sum \lambda_m$, the relative value of each λ_m is the fraction of the variance of the data represented by the associated eigenvector. If the eigenvalues are ordered by size (that is, by fraction of the variance explained by the corresponding eigenvector), it is usually found that only a limited number of eigenmodes are necessary to account for a very large fraction of the variance, say 3 to 5 (e.g., Kundu, et al., 1975; Servain and Legler, 1986; Fukumori and Wunsch, 1991). For many oceanographic applications it is satisfactory to retain only those first M eigenmodes that account for, say, 90 or 95% of the variance and subsequent profiles (the "synthetic profiles," as they were called in Carnes, et al., 1990) are constructed using only M eigenvectors.

If a given profile f has data at each of the K depths, estimation of the principal components is straightforward and is given by eq. (2) above. However, if the profile is incomplete, a modified procedure must be used. In this note we present one such procedure based upon least squares minimization.

2. Theory

According to eq. (1) above, a "synthetic profile" p may be constructed from M predetermined eigenvectors e_m according to the equations

$$p_k = T_k + \sum_m e_{km} c_m$$

where k is the depth index, $k = 1, \dots, K$. If all depths are present, eq. (2) shows the coefficients c_m are calculated from the relationship

$$c_m = \sum_k e_{mk} p'_k.$$

In practice, however, some of the K depth values may be missing: the profile may not start close enough to the surface, or it may have terminated at too shallow a depth or there may have been noise contamination within a certain depth range. If only an occasional value is missing, it may be reconstructed by interpolation and the expansion coefficients may then be computed from a complete data set. However, if interpolation is not satisfactory or feasible, a least-squares minimization approach may be used in which we determine those principle components c_m which minimize the error ϵ , the sum of the squares of the differences between a perturbation profile

$$p'_k = p_k - T_k$$

and its "synthetic" reconstruction:

$$\epsilon = \sum_k (p'_k - \sum_m c_m e_{km})^2.$$

This involves solving the system of equations

$$\begin{bmatrix} e_{k1} & e_{k1} & e_{k1} e_{k2} & \dots & e_{k1} e_{kM} \\ e_{k2} & e_{k1} & & & \\ \vdots & & & & \\ \vdots & & & & \\ e_{kM} & e_{k1} & \dots & \dots & e_{kM} e_{kM} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ \vdots \\ c_M \end{bmatrix} = \begin{bmatrix} p'_k e_{k1} \\ p'_k e_{k2} \\ \vdots \\ \vdots \\ p'_k e_{kM} \end{bmatrix}. \quad (3)$$

The resulting set of coefficients $\{c_m\}$ constitute the desired estimates of the true coefficients.

3. Example

As an example of the application of this technique, consider the 28 temperature profiles shown in Fig. 1a. Table 1 lists the measurement depths of the profiles. Fig. 1b shows the mean temperature profile. The data represent a set of CTD temperature profiles taken in July during an experiment in the Sargasso Sea, but any other comparable data set would do as well for describing our technique. Shown in Fig. 2 are the first five EOF modes (i.e., e_1 through e_5), which jointly explain 98% of the variance apparent in Fig. 1c. The first five principal components for each perturbation profile in Fig. 1c as estimated from eq. (2) are given in Fig. 3a, and the root-mean-square (RMS) difference between the true profiles and the synthetic profiles generated with these five principal components and the five eigenfunctions in Fig. 2 is given in Fig. 3b. Using the five eigenfunctions, the maximum RMS deviation is about 0.25 °C near 50 m, which is located somewhat below the base of

the mixed layer at around 30 m. Otherwise the mean deviation is typically less than 0.15 °C.

First consider the impact on the coefficient values of having missing data near the surface. We estimated the principal components for six separate cases with increasingly large amounts of missing data, as described in Table 2. The resulting coefficients, normalized by their true values computed from the complete profiles (Fig. 3a), are shown in Fig. 4, and the RMS differences between the true profiles and their corresponding synthetic profiles constructed from five modes are given in Fig. 5. Not surprisingly, as the range of the missing data points becomes larger, in general the deviation of the estimated coefficients from their true values becomes larger, as does the RMS difference across the depth range of the missing data. However, the coefficients of certain modes are much less sensitive than the coefficients of other modes.

The coefficients of the first mode are least sensitive to data deletion, but by the time the first 100 m of data are removed (Case 5), the individual coefficients begin to deviate noticeably from their true values. Mode 4 coefficients were the most sensitive, followed by mode 5, then mode 3, and mode 2. This correlates well in a qualitative sense with the location of extrema in the eigenfunctions of Fig. 2. However, even with the upper 100 (Case 5) to 150 (Case 6) m of data removed, the RMS deviation is less than 1.7 °C. Notice that the RMS difference below the depth of the missing data actually is less than it is when using the full profiles. This is because when complete profiles are used, a trade-off must occur in the calculation of the principal components, where the various eigenfunctions must be fit as well as possible over the full depth range. When the highly structured upper part of the profile is missing, the remaining portion may be better fit with the eigenfunctions.

Next consider what occurs if increasing amounts of data are missing at the bottom of the profiles. The six cases considered are described in Table 3, the normalized coefficients are in Fig. 6, and the RMS differences are in Fig. 7. Because there is so much less structure at the bottom of the profiles, the impact of missing data is less deleterious. In general, as the amount of missing data becomes larger, the deviation of the estimated coefficients from their true values also becomes larger. However, the deviations are much smaller in comparison with the estimated coefficients constructed with near-surface data points removed. RMS differences are also much smaller; less than 0.3 °C in Case 12 when 1400 m of data were removed from the bottom of each profile.

Again the coefficients of the first mode are by far the least sensitive to the deletion of data; they do not vary significantly from their true values even when a large number of near-bottom data points are missing from the profiles. Modes 2 and 5 are the most sensitive, but their deviations from their respective true values are much less than when the surface values are removed. Deviations in mode 3 increase quite slowly as the amount of missing data increases.

The RMS difference increases as the profiles are progressively truncated. The most

sensitive part of the profiles lies between about 600 to 1200 m, where mode 2 in particular has considerable structure. Deletion of data below this depth range has only a modest impact upon the resulting synthetic profiles. The RMS difference above the truncation depths again tends to be somewhat less than the RMS difference computed using full depth profiles.

Finally, we deleted increasingly larger amounts of data in the depth ranges centered around 300 m (a minimum in the variability seen in Fig. 1c) and around 600 m (a maximum in the variability), as described in Tables 4 (Cases 13, 14, and 15) and 5 (Cases 16, 17, and 18). The normalized coefficients and RMS differences are given in Figs. 8 and 9 and Figs. 10 and 11. Both of these sets of results show deletion of data increases the RMS differences between the synthetic profiles and the data over that computed with complete data; however, the increase in the RMS is only in the vicinity of the missing data and it is remarkably small. Case 18 was computed with data between 500 - 700 m deleted, but the RMS difference at 500 m increased less than 0.1 °C to about 0.2 °C.

4. Coefficient Estimation Software

To implement the theory described above, a FORTRAN subroutine ESTCOE was written and is presented in Appendix A. ESTCOE requires the input of two data files; one containing the profiles themselves and the other containing the mean profile and eigenvectors associated with the area of interest. For each profile, the system of equations given in eq. (3) is solved for the set of unknown estimated coefficients $\{c_m\}$. The synthetic profile is then constructed using the estimated coefficients, the input eigenvectors, and the mean profile.

The output coefficients are stored in a two-dimensional variable, Coef(MaxProfiles, Maxz), where the maximum number of profiles and maximum number of depths have been set (by a parameter statement) as MaxProfiles = 100 and Maxz = 50, respectively. The output synthetic profiles are also stored in a two-dimensional variable, pp(MaxProfiles, Maxz).

Other parameters are returned from ESTCOE. A full description of these parameters as well as a more complete description of the algorithm, format of input files, output parameters, an example driver program, and the FORTRAN code itself is found in Appendix A.

5. Concluding Remarks

We have described a least-squares minimization technique for estimating principal components from incomplete ocean profile data. An important application of this technique is to compute from these estimated principal components a synthetic profile that interpolates or extrapolates the missing data values. This synthetic profile can then be used in place of the original incomplete profile, or it can be merged with the original profile in some fashion to fill in the missing data. In particular, the technique lends itself to the problem of

extrapolating shallow ocean data profiles -- say, from shallow CTD casts or from expendable probes such as XBTs (expendable bathythermographs) or AXBTs (the air-launched XBT) - - to a deeper preselected depth. A commonly used alternative is to use climatological values below the maximum data depth, with some merging procedure to combine the two. The technique presented here, however, makes use of the information contained in the structure of the observed profile and the correlations between that upper level structure and the deeper structure to produce what will, in many cases, be a better estimate of the deeper structure than can be made with climatology alone.

To illustrate this, we computed the RMS difference between our 28 temperature profiles in Fig. 1a and the appropriate GDEM (Teague, et al., 1990) and Levitus (Levitus, 1982) climatological profiles (Fig. 12). We also simulated the use of T-7 XBTs by cutting our 28 profiles off below 700 m, estimating the principal components as described above, and computing the synthetic profiles from these principal components (this is Case 10 of Fig. 7). The RMS difference for Case 10 between the synthetic profiles and the data is also reproduced on Fig. 12. The RMS difference for the synthetic profiles is substantially lower than the RMS differences for either of the climatologies, particularly below 700 m. Hence, merging the synthetic profiles with the 700 m deep measured profiles would give a better depth-extended data set than using either climatology. These profiles happened to come from the Sargasso Sea region, but we feel confident our results would hold elsewhere as well.

6. References

- Carnes, M. R., J. L. Mitchell and P. W. DeWitt (1990). Synthetic temperature profiles derived from Geosat altimetry: Comparison with air-dropped expendable bathythermograph profiles. *J. Geophys. Res.* 95: 17979-17992.
- Davis, R. E. (1976). Predictability of sea surface temperatures and sea level pressure anomalies over the North Pacific Ocean. *J. Phys. Oceanogr.* 6: 249-266.
- Fukumori, I., and C. Wunsch (1991). Efficient representation of the North Atlantic hydrographic and chemical distributions. *Prog. Oceanogr.* 27: 111-195.
- Halliwell, G. R., Jr., and C. N. K. Mooers (1979). The space-time structure and variability of the shelf water-slope water and Gulf Stream surface temperature fronts and associated warm core eddies. *J. Geophys. Res.*: 84, 7707-7725.
- Kundu, P. K., J. S. Allen, and R. L. Smith (1975). Modal decomposition of the velocity field near the Oregon coast. *J. Phys. Oceanogr.* 5: 683-704.
- Lagerloef, G. S. E., and R. L. Bernstein (1988). Empirical orthogonal function analysis of advanced very high resolution radiometer surface temperature patterns in Santa Barbara Channel. *J. Geophys. Res.* 93: 6863-6873.
- Levitus, S. (1982). Climatological atlas of the world ocean. *NOAA Tech. Pap.*, 3, 173 pp.
- Moore, D. (1974). Empirical orthogonal functions -- A non-statistical view. MODE Hot Line News, Oct. 25 1974, 1-6.
- Servain, J., and D. M. Legler (1986). Empirical orthogonal function analysis of tropical Atlantic sea surface temperature and wind stress: 1964-1979. *J. Geophys. Res.* 91: 14181-14191.
- Teague, W. J., M. J. Carron and P. J. Hogan (1990). A comparison between the generalized digital environmental model and Levitus climatologies. *J. Geophys. Res.* 95: 7167-7183.
- Tolstoy, A., and O. Diachok (1991). Acoustic tomography via matched field processing. *J. Acoust. Soc. Am.* 89(3): 1119-1127.

Table 1. Standard depths (m) used in the analysis.

0	10	20	30	50	75	100
125	150	200	250	300	400	500
600	700	800	900	1000	1100	1200
1300	1400	1500	1750	2000		

Table 2. Depths at which data values were eliminated for Cases 1 - 6.

- Case 1: 0 m
- Case 2: 0, 10, 20 m
- Case 3: 0, 10, 20, 30 m
- Case 4: 0, 10, 20, 30, 50 m
- Case 5: 0, 10, 20, 30, 50, 75, 100 m
- Case 6: 0, 10, 20, 30, 50, 75, 100, 125, 150 m

Table 3. Depths at which data values were eliminated for Cases 7 - 12.

- Case 7: 2000, 1750 m
- Case 8: 2000, 1750, 1500, 1400, 1300, 1200, 1100 m
- Case 9: 2000, 1750, 1500, 1400, 1300, 1200, 1100, 1000, 900 m
- Case 10: 2000, 1750, 1500, 1400, 1300, 1200, 1100, 1000, 900, 800 m
- Case 11: 2000, 1750, 1500, 1400, 1300, 1200, 1100, 1000, 900, 800, 700 m
- Case 12: 2000, 1750, 1500, 1400, 1300, 1200, 1100, 1000, 900, 800, 700, 600 m

Table 4. Depths at which data values were eliminated for Cases 13 - 15.

- Case 13: 300 m
- Case 14: 250, 300 m
- Case 15: 200, 250, 300, 400 m

Table 5. Depths at which data values were eliminated for Cases 16 - 18.

- Case 16: 600 m
- Case 17: 500, 600 m
- Case 18: 500, 600, 700 m

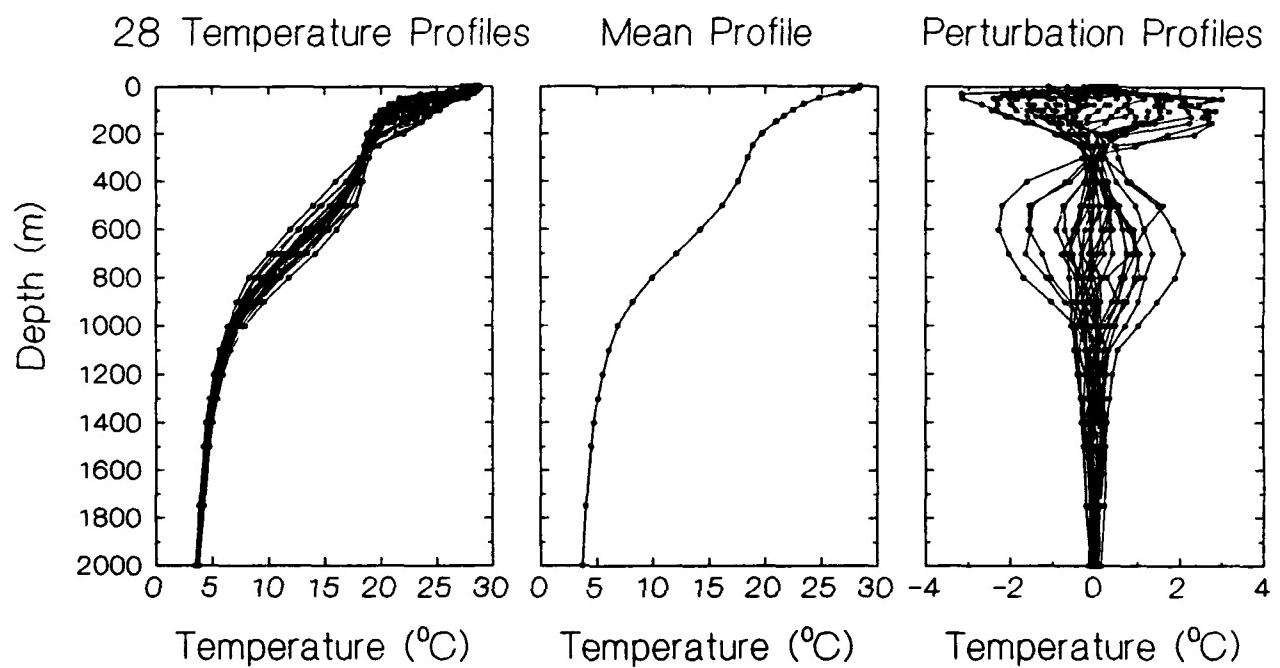


Figure 1. The 28 temperature profiles used in the analysis. a: overplot of all profiles; b: mean profile; c: overplot of perturbation profiles.

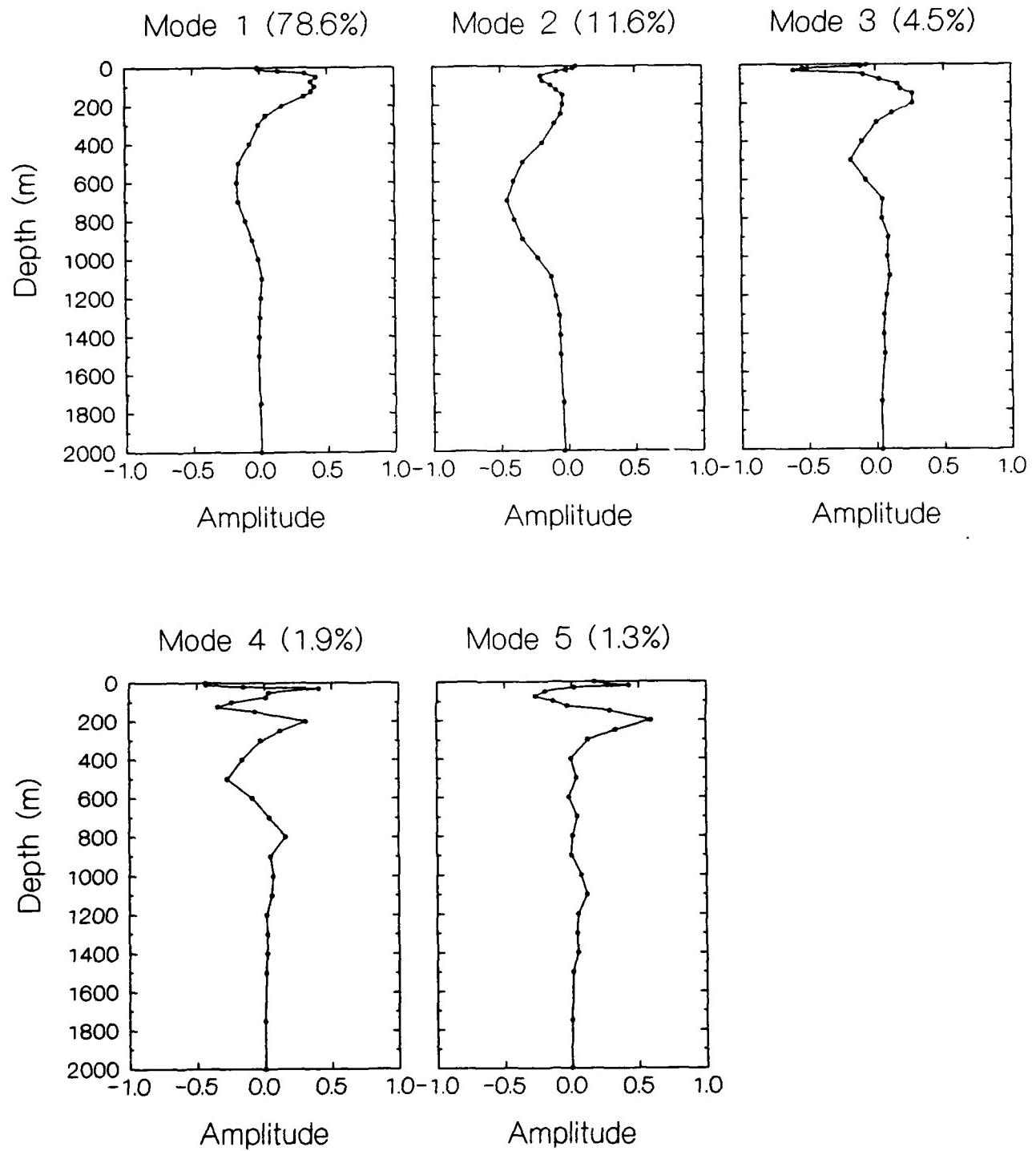


Figure 2. The first five eigenvectors computed from the data in Fig. 1. The number in parentheses is the percent of the total variance that is described by that particular eigenvector.

First 5 principal components
for 28 temperature profiles

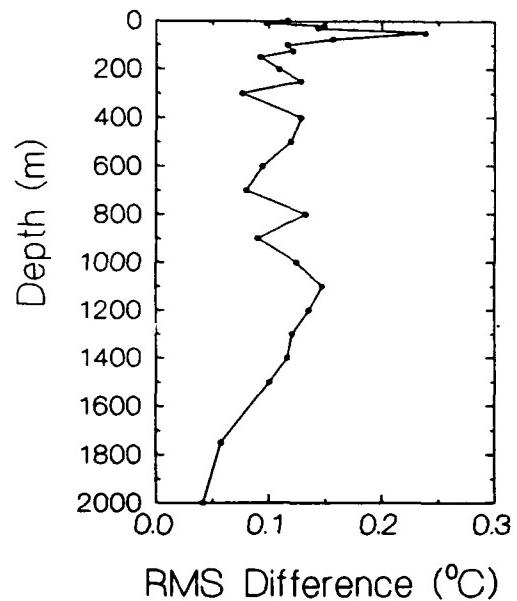
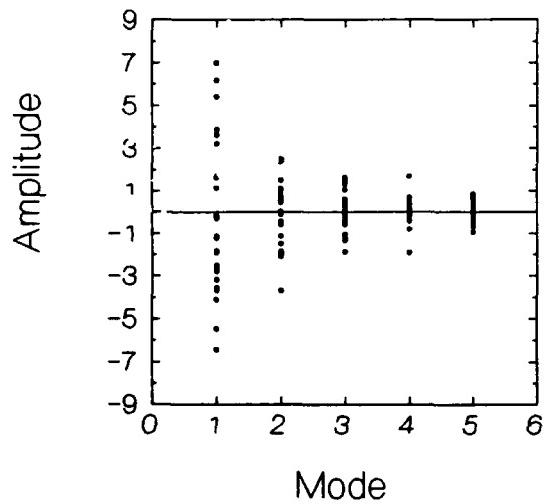


Figure 3. a: The expansion coefficient values versus mode number for coefficients calculated using full depth profiles; b: the RMS temperature difference over all 28 profiles between the real and the synthetic profiles computed with five modes.

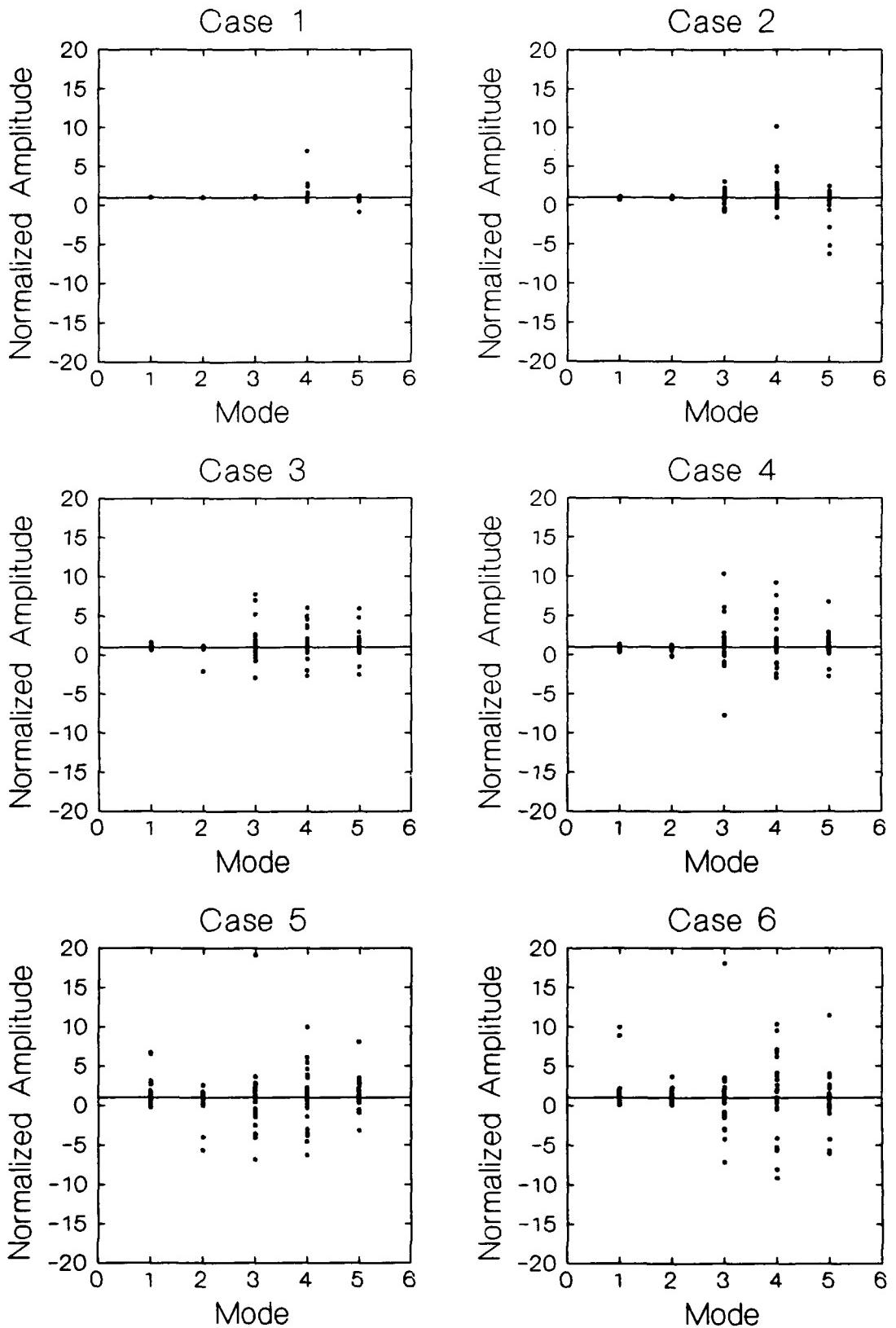


Figure 4. The normalized estimated coefficient amplitudes versus mode number for Cases 1 - 6 in which data in the upper portion of the profiles were deleted. Table 2 describes the data deleted for each case.

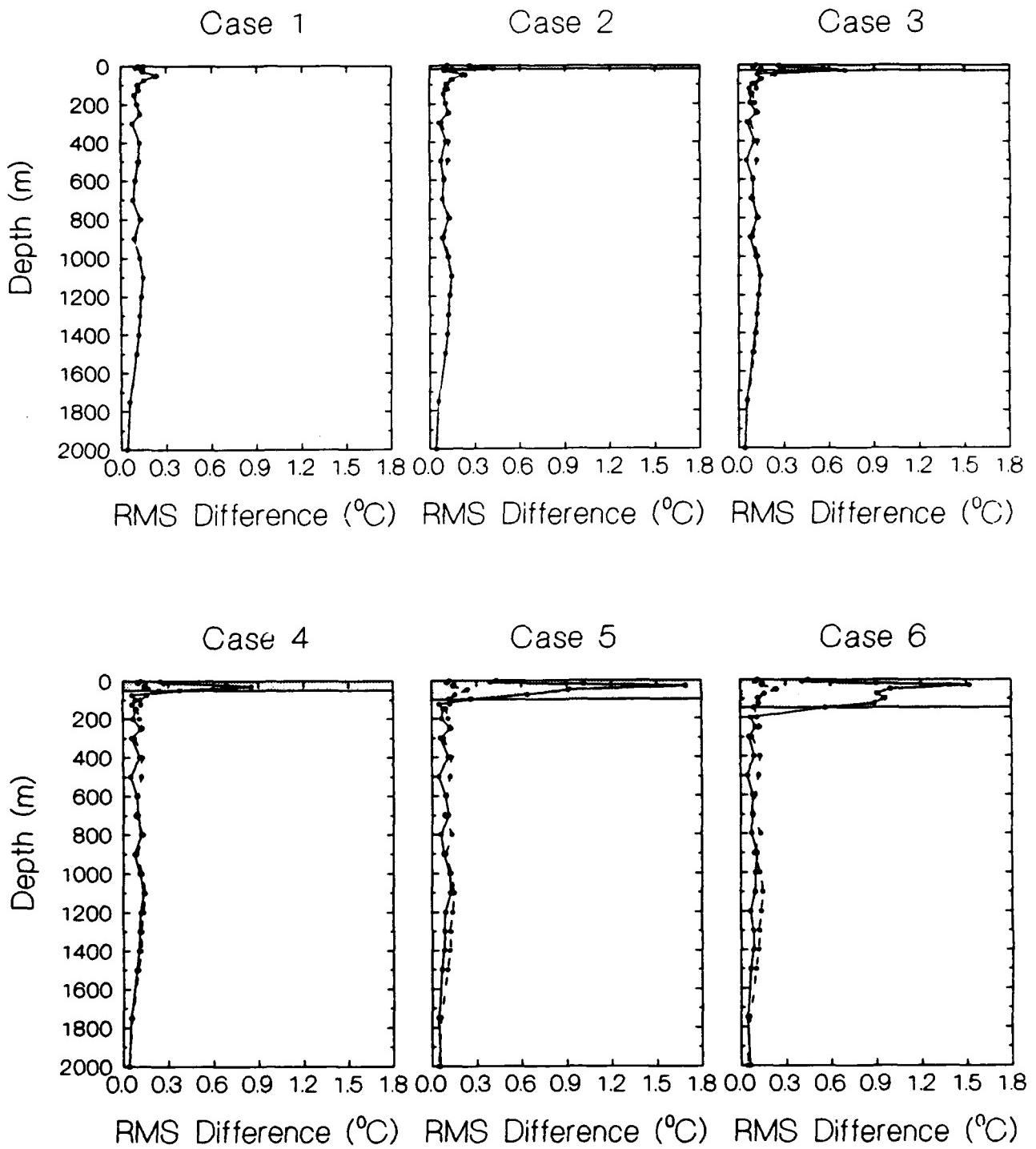


Figure 5. The RMS differences for Cases 1 - 6 between the real and the synthetic profiles computed using the estimated coefficients given in Fig. 4. The dotted curve is the RMS difference from Fig. 3b, shown for comparison. The shaded regions indicate the depth range over which data was deleted.

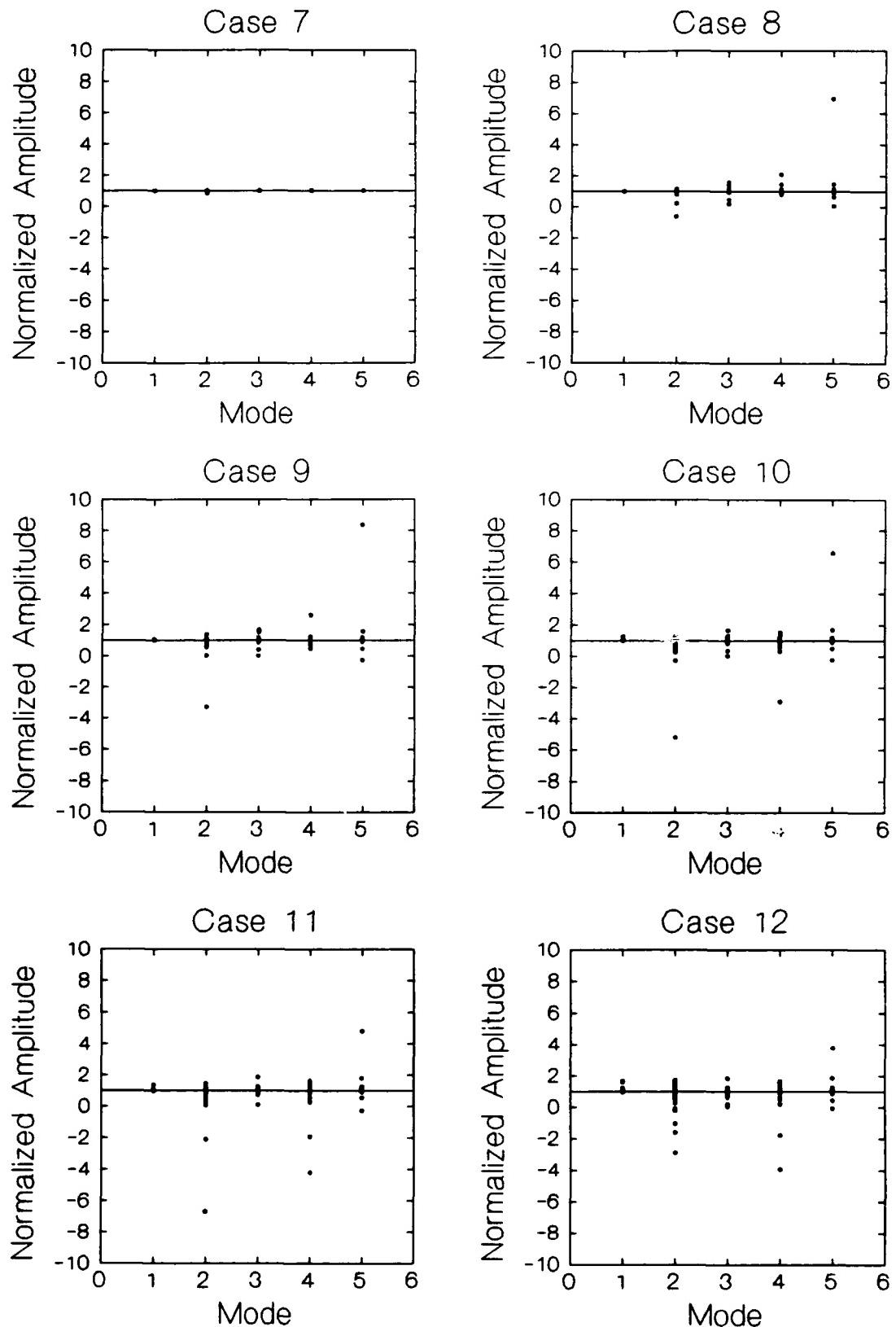


Figure 6. The normalized estimated coefficient amplitudes versus mode number for Cases 7 - 12 in which data in the lower portion of the profiles were deleted. Table 3 describes the data deleted for each case.

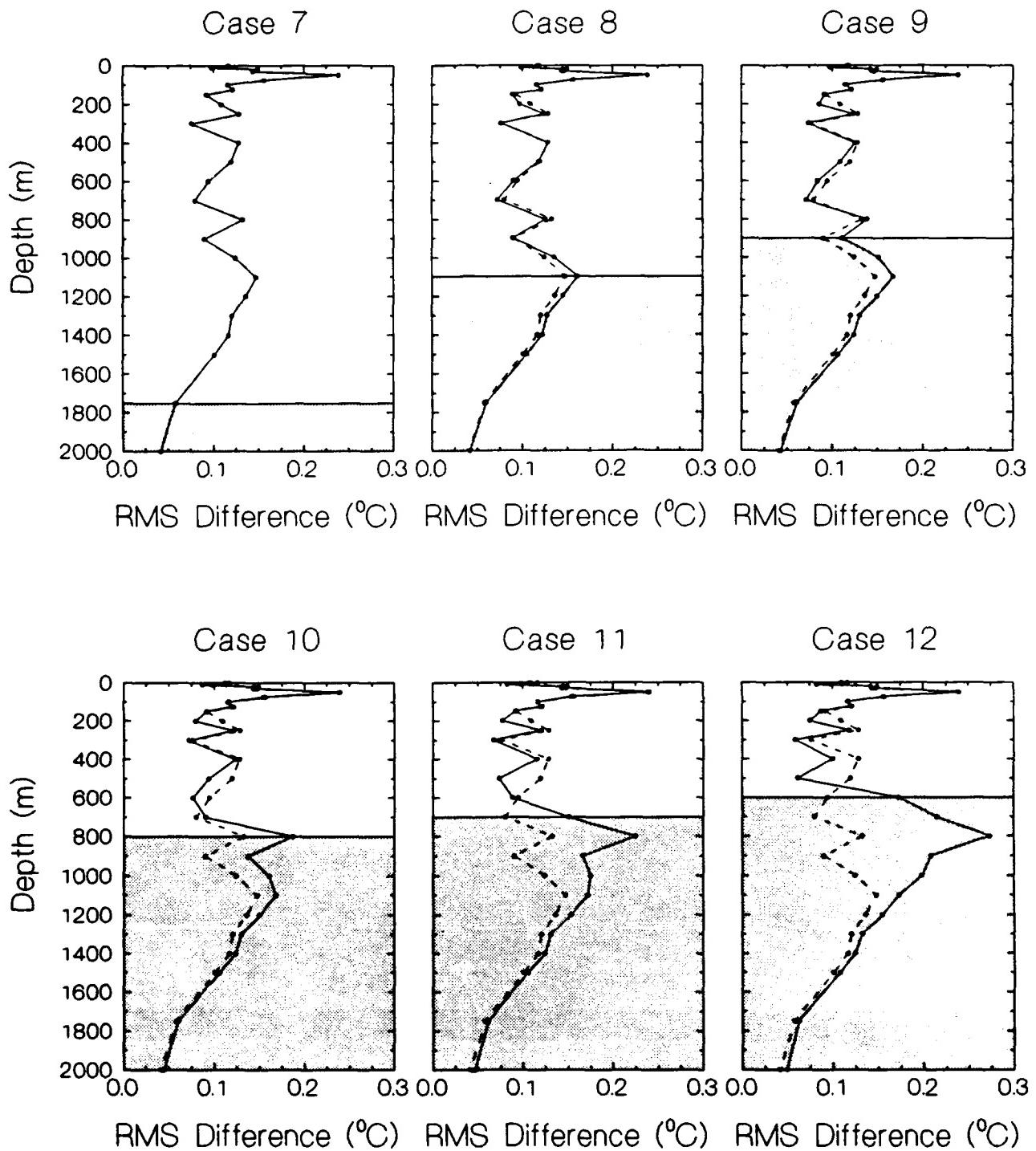


Figure 7. The RMS differences for Cases 7 - 12 between the real and the synthetic profiles computed using the estimated coefficients given in Fig. 6. The dotted curve is the RMS difference from Fig. 3b, shown for comparison. The shaded regions indicate the depth range over which data was deleted.

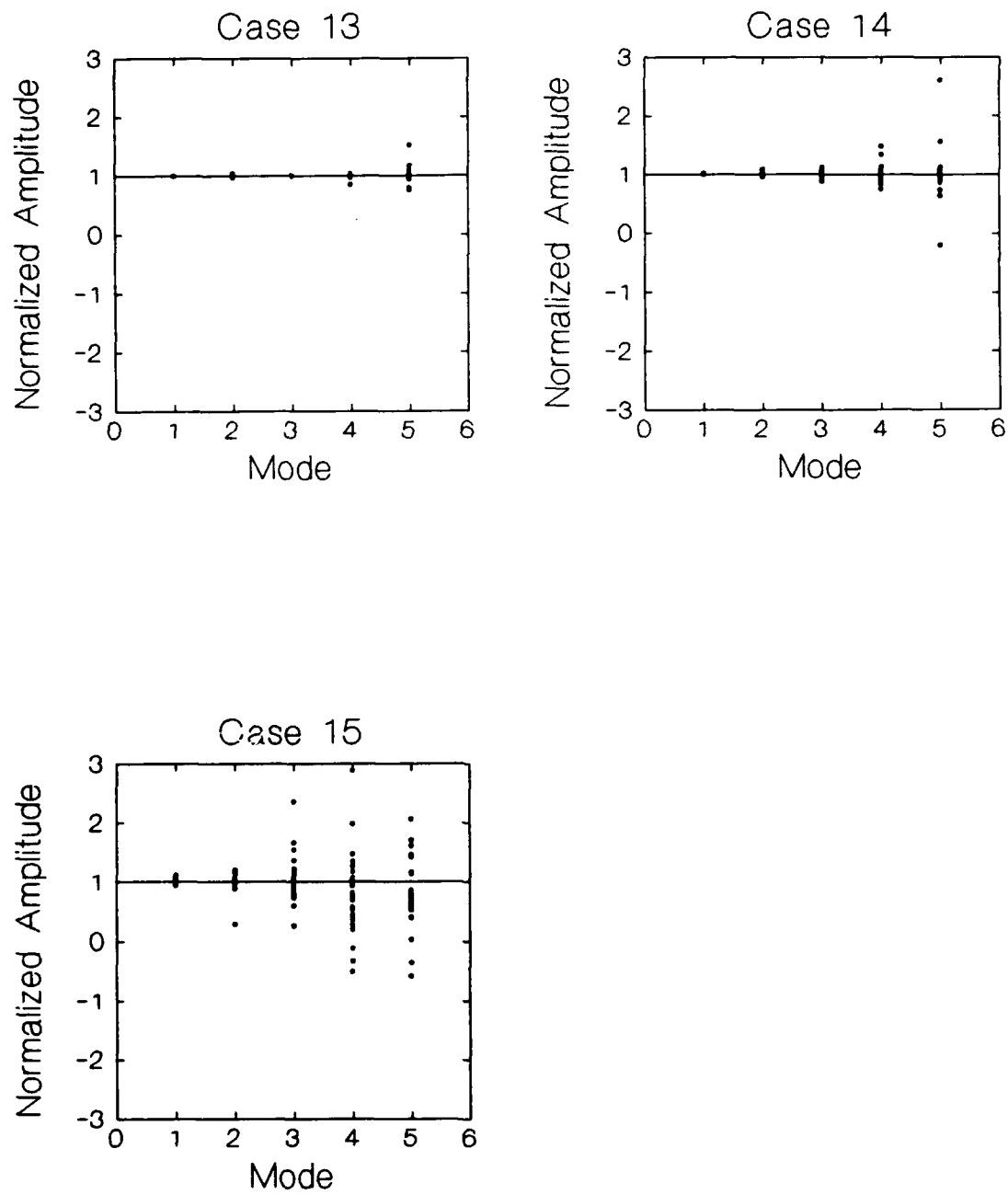


Figure 8. The normalized estimated coefficient amplitudes versus mode number for Cases 13 - 15 in which data around 300 m (a data variance minimum) were deleted. Table 4 describes the data deleted for each case.

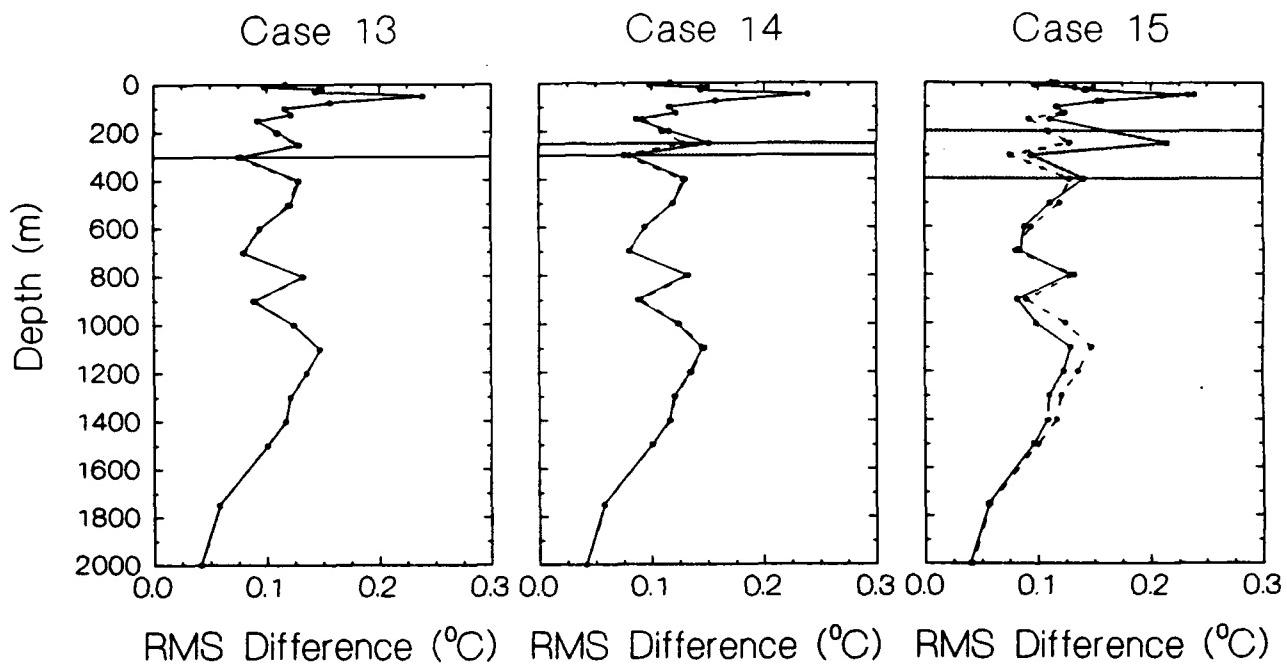


Figure 9. The RMS differences for Cases 13 - 15 between the real and the synthetic profiles computed using the estimated coefficients given in Fig. 8. The dotted curve is the RMS difference from Fig. 3b, shown for comparison. The shaded regions indicate the depth range over which data was deleted.

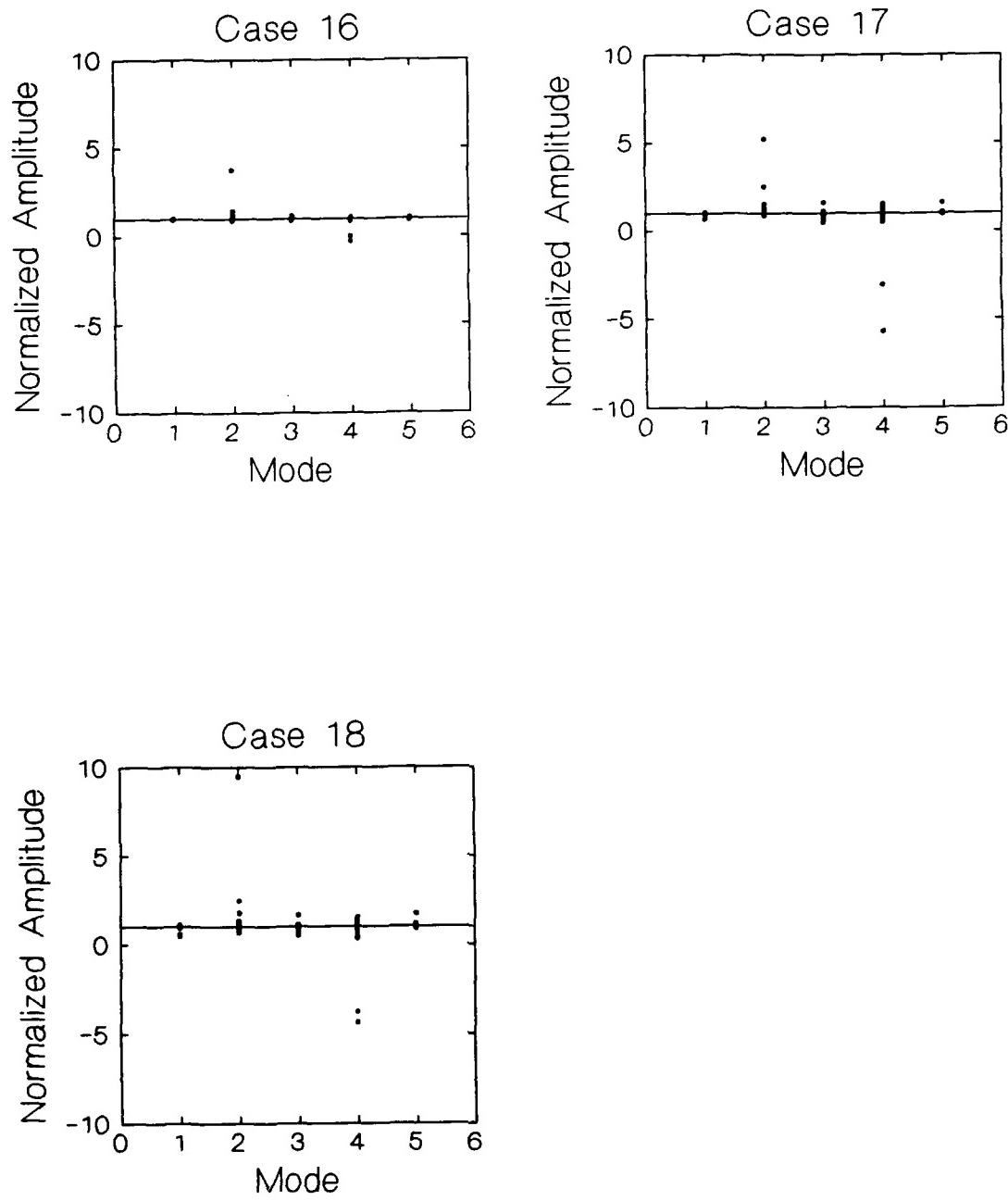


Figure 10. The normalized estimated coefficient amplitudes versus mode number for Cases 16 - 18 in which data around 600 m (a data variance maximum) were deleted. Table 5 describes the data deleted for each case.

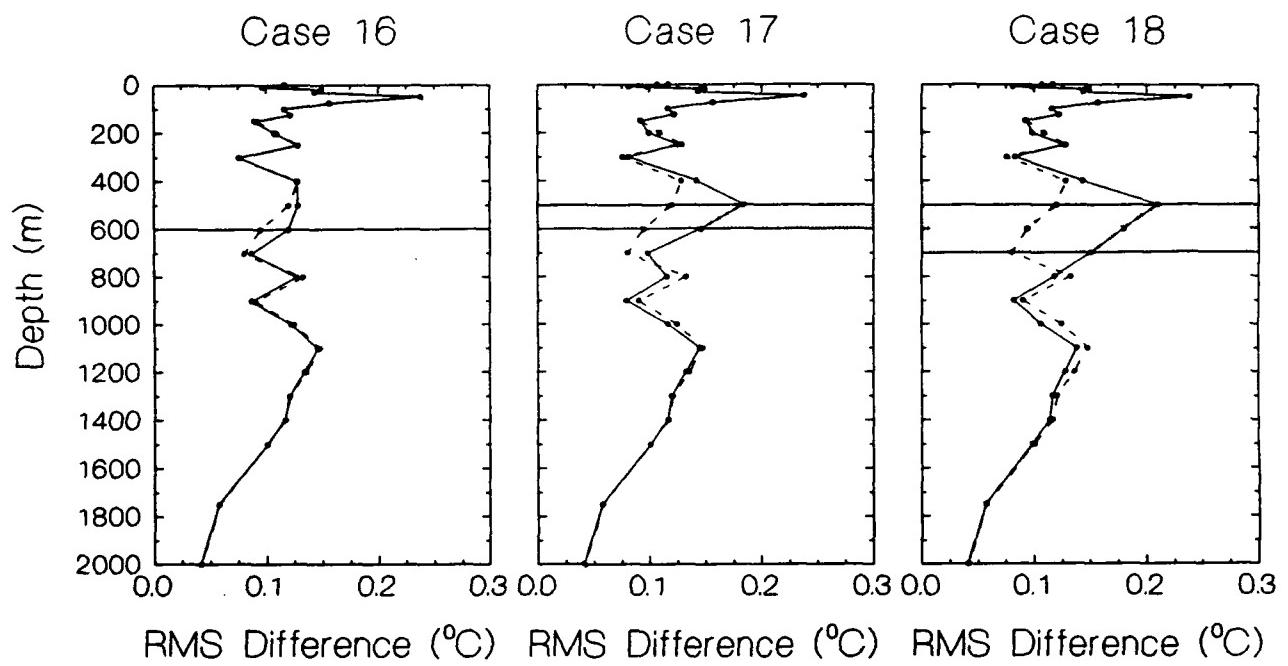


Figure 11. The RMS differences for Cases 16 - 18 between the real and the synthetic profiles computed using the estimated coefficients given in Fig. 10. The dotted curve is the RMS difference from Fig. 3b, shown for comparison. The shaded regions indicate the depth range over which data was deleted.

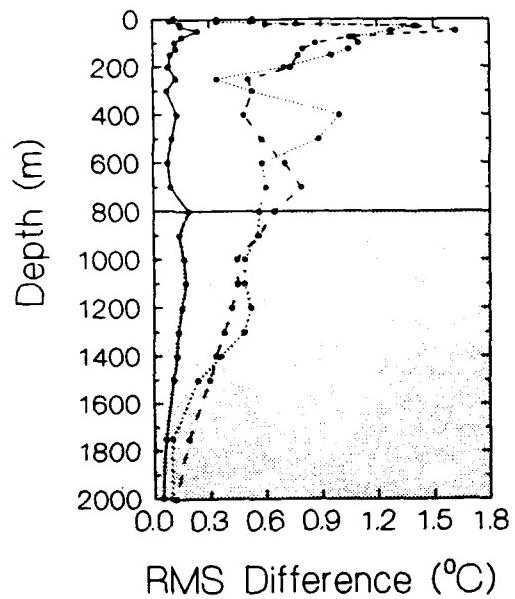


Figure 12. RMS differences between the data in Fig. 1a and GDEM climatology (dotted line), Levitus climatology (dashed line) and synthetic profiles computed from data truncated below 700 m to simulate T-7 XBTs (solid line). At all depths the synthetic profiles exhibit a smaller RMS difference and hence provide a better estimate of the profiles below 700 m than either of the climatologies.

Appendix A
FORTRAN Subprogram

FORTRAN Subroutine ESTCOE

I. Introduction

Subroutine ESTCOE estimates EOF coefficients for a set of input data profiles. It also calculates synthetic profiles using the estimated coefficients.

II. Input

ESTCOE requires the unit numbers of 2 input files:

- inunit1: The mean profile & eigenvectors.
- inunit2: The data profiles themselves.

The mean profile & eigenvectors must be arranged in the first input file as follows:

Depth values (of the mean profile) are stored in column 1, and the mean profile values at their respective depths must be stored in column 2. An "end of record" flag is required at the end of the mean profile. This end of record flag is a pair of real numbers, 9999.00 9999.00.

After the mean profile end of record flag, the eigenvectors are stored sequentially. For each eigenvector, the depth values should be stored in column 1, and the eigenvector values at their respective depths must appear in column 2. An end of record flag is required at the end of each eigenvector "record."

NOTE: The number of depths in the mean profile/eigenvector file has been set to 50 via a parameter statement: parameter(Maxz = 50). This is also the maximum number of eigenvectors allowed in the mean profile/eigenvector file.

The data profiles must be arranged in the second input file as follows:

Depth values are stored in column 1, and the data values at their corresponding depths are stored in column 2. Depth values should be sequential; that is, a measurement taken at depth 10 m should be stored before a measurement taken at 20 m. An end of record flag is required at the end of each profile.

NOTE: The number of depths in any single profile has been set to 6000 via a parameter statement: parameter(Maxpts = 6000).

III. Processing

First, the mean profile & eigenvector input file is read. The user is asked to decide how many modes will be used to build the coefficients (and subsequent synthetic profiles).
NOTE: The maximum number of modes available is equal to the number of eigenvectors

in this input file. Then, one at a time, the profiles are read, coefficients are calculated, and synthetic profiles are constructed using the following approach:

- a. When a profile is loaded, it is checked against the eigenvector depths for missing data points (relative to the eigenvector depths). If a profile contains data at depths that are not in the eigenvector file, these "extra depths" are ignored in subsequent coefficient estimation subroutines. If a profile contains "missing data" relative to the eigenvector file, this is noted in a "flag" array.
- b. For a single profile, the system of equations given in eq. (3) is solved using standard Gauss-Jordan elimination routines¹.
- c. The eigenvector matrix $[e_1 e_2 \dots e_K]$ is created using only the eigenvector depths where input data values are present. To solve for the coefficient vector represented by the set of values $\{c_m\}$, the inverse of the eigenvector matrix is found. The determinant of the resulting matrix is displayed. If the determinant is too small (relative to a set tolerance of 1.0D-14), the user is warned that there may be large errors in the resulting estimated coefficients. The user is encouraged to rerun the program using a smaller number of modes with which to build the coefficients.
- d. The right-hand side vector in eq. (3) is created using the input profile data, the mean profile, and the eigenvectors. The entire system of equations is then solved.
- e. The synthetic profile is built using the estimated coefficients. The coefficients and the synthetic profile are stored, and the program "loops" through again using the next profile's data.

IV. Output

Output from ESTCOE consists of the variables listed below. Parameter values are set for the maximum number of depths allowed in the mean profile/eigenvector input file (Maxz = 50) and the maximum number of input profiles (MaxProfiles = 100).

Num_probes:	Number of profiles in the input data file
e_n_dep:	Number of depths in each eigenvector (as well as the mean profile)
Num_eigen:	Number of eigenvectors (or modes) used in calculating the estimated coefficients
e_depth:	Array of eigenvector depths; e_depth(MaxDepths)

¹Press, William H. et al. (1986). *Numerical Recipes -- The Art of Scientific Computing*. New York (NY): Cambridge University Press, pp. 35-37.

Coef: 2-D array that stores the estimated coefficients;
Coef(MaxProfiles, Maxz)
pp: 2-D array that stores the synthetic profiles; pp(MaxProfiles,
Maxz)

V. Example of input files and driver program

(1) Input file 1: Mean temperature profile and eigenvectors

0.00	28.38	(depth 0 m, mean profile value at 0 m)
10.00	28.33	
...	...	
1750.00	4.03	
2000.00	3.66	(depth 2000 m , mean profile value at 2000 m)
9999.00	9999.00	(end of record)
0.00	-0.01	(depth 0 m, eigenvector 1)
10.00	0.04	
...	...	
1750.00	0.10	
2000.00	0.04	(depth 2000 m, eigenvector 1)
9999.00	9999.00	(end of record)
...	...	
0.00	0.06	(depth 0 m, eigenvector N)
10.00	0.04	
...	...	
1750.00	-0.19	
2000.00	0.00	(depth 2000 m, eigenvector N)
9999.00	9999.00	(end of record)

(2) Input file 2: Temperature profiles

0.00	28.60	(depth 0 m, temperature value at 0 m, profile 1)
2.00	28.60	
4.00	28.58	
5.00	28.57	
8.00	28.40	
10.00	28.24	
...	...	
1750.00	3.96	
9999.00	9999.00	(end of record)
...	...	
0.00	28.29	(depth 0 m, temperature value at 0 m, profile N)
5.00	28.28	
10.00	28.27	
...	...	
1700.00	3.81	
9999.00	9999.00	(end of record)

(3) Example of a driver program that calls the subroutine ESTCOE.

```
$large
program driver
```

c Purpose: To test the subroutine ESTCOE

c Compiler: Microsoft FORTRAN v 4.01

c To compile: f1 -c driver.for

c To link: link driver + ESTCOE;

c To run: driver

c Parameter values set: Maximum number of depths in any input file is set at Maxz = 50

c Maximum number of input profiles is set at MaxProfiles = 100

```
parameter(Maxz = 50, MaxProfiles = 100)
double precision Coef(MaxProfiles, Maxz), pp(MaxProfiles, Maxz)
integer inun1, inun2, outun1, outun2
integer Num_eigen, Num_probes, e_n_dep, mode, probe
real e_depth(Maxz)
```

c Set unit numbers for input/output files

inun1 = 10

inun2 = 11

outun1 = 12

outun2 = 13

c Give input filenames

open(inun1, file = 'evec.dat')

open(inun2, file = 'profles.dat')

c Give the output filenames

open(outun1, file = 'coeff.dat')

open(outun2, file = 'synthet.dat')

c Call the estimate coefficient subroutine

```
call est_coef(inun1, inun2, Num_probes, e_n_dep, Num_eigen,
*           e_depth, Coef, pp)
```

c Write the coefficients to output file1

4 format(1x, i4, 1x, f10.4)

do 10 probe = 1, Num_probes

do 20 mode = 1, Num_eigen

write(outun1, 4) mode, Coef(probe, mode)

```
20    continue
10    continue

c  Write the synthetic profiles to output file2
5   format(1x, f10.2, 1x, f10.4)
     do 30 probe = 1, Num_probes
         do 40 k = 1, e_n_dep
             write(outun2, 5) e_depth(k), pp(probe, k)
40    continue
30    continue

c  Close all open files
close(inun1)
close(inun2)
close(outun1)
close(outun2)

c  End of program
stop
end
```

```

$large
c **** subroutine ESTCOE(inun1, inun2, Num_probes, e_n_dep,
*           Num_eigen, e_depth, Coef, pp)
c ****
c
c Purpose: To find EOF coefficients for input profiles which
c have missing depth values and to create synthetic
c profiles.
c
c Input:   inun1: Unit number of input profiles
c          inun2: Unit number of Mean profile & Eigenvectors
c
c Output:  Num_probes: Number of profiles in the input data
c          file
c          e_n_dep: Number of depths in each eigenvector (as
c                     well as the mean profile)
c          Num_eigen: Number of eigenvectors (or modes) used
c                     in calculating the estimated
c                     coefficients
c          e_depth: Array of eigenvector depths;
c                     e_depth(Maxz)
c          Coef:    2-D array which stores the estimated
c                     coefficients; Coef(MaxProfiles, Maxz)
c          pp:     2-D array which stores the synthetic
c                     profiles; pp(MaxProfiles, Maxz)
c
c Limitations: Maximum number of input profiles is set at
c               MaxProfiles = 100. Maximum number of depths in any
c               single input profile is set at Maxpts = 6000.
c               Maximum number of depths in the mean/eigenvector file
c               is set at Maxz = 50. Matrix Singularity criteria set
c               at det_tol = 1.0D-14.
c
c Precision: Double Precision
c Compiler: Microsoft FORTRAN v 4.01
c To compile: f1 -c ESTCOE.for
c To link: link driver + ESTCOE;
c
c Code uses routines from NUMERICAL RECIPES
c ROUTINES: L U Decomposition (LUDCMP)
c            Back Substitution (LUBKSB)
c NOTE: Modifications were made to the original code for these two
c       subroutines; real variables were changed to double precision;
c       unnecessary output variables were deleted from calling and
c       subroutine declaration statements.

```

```

c
c Program written by: Eileen P. Kennelly
c                      Neptune Sciences, Inc
c                      Slidell, LA 70458
c Variables:
c A:      "A" matrix in A * x = B; Used in LUDCMP
c B:      "B" vector in A * x = B; x = (inv)A * B; The
c         coefficients for a particular profile are temporarily
c         stored in this vector; Used in LUBKSB
c Coef:    2-D array which stores calculated coefficients;
c           Coef(MaxProfiles, Maxz)
c determin: Determinant of the LUDCMP matrix; checks for
c singularity
c det_tol: Tolerance check of the determinant of the LUDCMP
c           output matrix; set at parameter (det_tol = 1.0D-14)
c E:      Eigenvectors are stored in this 2-D matrix;
c           E(Maxz, Maxz)
c eof:    End of input file flag
c e_depth: Array of eigenvector depths; e_depth(Maxz)
c e_n_dep: Number of depths in each eigenvector
c f:      Shallow water data (input profiles) is stored here
c         one profile at a time; a 1-D array, f(Maxpts)
c fbar:   Mean profile associated with the eigenvectors; a 1-D
c         array fbar(Maxz)
c flag:   Logical array; Set to TRUE if there exists an
c         eigenvector value at a profile depth
c f_depth: Array of profile depths; f_depth(Maxpts)
c f_n_dep: Number of depths in a particular profile
c f_prime: At depth z, f' = Profile value - Mean;
c           f'(z) = f(z) - fbar(z); a 1D array f_prime(Maxpts)
c indx:   Array used by LUDCMP and LUBKSB
c inun1:  Unit number of input Mean profile and eigenvectors
c inun2:  Unit number of input profiles
c MaxProfiles: Maximum number of input profiles;
c               parameter (MaxProfiles = 100)
c Maxz:   Maximum number of depths in either input file;
c               parameter (Maxz = 50)
c N:      Number of eigenvectors in the eigenvector file
c Num_eigen: Number of eigenvectors which are actually used in the
c             construction of estimated coefficients
c Num_probes: Number of profiles in input file
c p:      1-D array which stores a single synthetic profile;
c           p(Maxz)
c pp:    2-D array which stores all synthetic profiles;
c           pp(MaxProfiles, Max)
c probe:  Profile counter

```

```

c ****
double precision det_tol
parameter (Maxz = 50, MaxProfiles = 100, Maxpts = 6000)
parameter (det_tol = 1.0D-14)
double precision A(Maxz, Maxz), B(Maxz), f(Maxpts), determin
double precision E(Maxz, Maxz), fbar(Maxz), f_prime(Maxpts)
double precision p(Maxz), Coef(MaxProfiles, Maxz)
double precision pp(MaxProfiles, Maxz)
integer inun1, inun2, indx(Maxz)
integer Num_eigen, probe, e_n_dep, f_n_dep, N, Num_probes
logical eof, flag(Maxz)
real e_depth(Maxz), f_depth(Maxpts)

c Get the Eigenvectors and the Mean vector from first input file
call EigMea(inun1, N, E, e_n_dep, e_depth, fbar)
call Ld_Eig(N, Num_eigen)

do 10 probe = 1, MaxProfiles

c Let user know something is happening...
write(*,*) 'Calculating coefficients for profile ', probe

c Load an input profile, f
call Ld_Pro(inun2, f, f_depth, f_n_dep, eof)
if (eof) goto 89

c Find missing depths and set the flag array to indicate this
call Miss(f_depth, f_n_dep, e_depth, e_n_dep, flag)

c Load the matrix A (in A * x = B) with the Eigenvectors
call A_Matx(e_n_dep, E, Num_eigen, flag, A)

c Call the Numerical Recipe LUDCMP to find the inverse of the A
c matrix in A * x = B
call LUDCMP(Num_eigen, A, indx)

c Look at the determinant of the inverse A matrix. Check it for
c singularity
call Det(Num_eigen, A, determin)
if (dabs(determin) .lt. det_tol) then
    call Message()
endif

c Calculate f_prime; f'(z) = f(z) - fbar(z)
call Ld_fpr(f, f_depth, f_n_dep, e_depth, e_n_dep,

```

```

*      fbar, f_prime)

c      Load vector B (in A * x = B) with f' * E
      call Ld_B(E, e_depth, e_n_dep, f_depth, f_n_dep,
*              f_prime, Num_eigen, B)

c      Solve for x: A * x = B
c      (resulting coefficients stored in variable B)
      call LUBKSB(A, Num_eigen, indx, B)

c      Create the synthetic profile,
      call Syn(B, E, fbar, e_n_dep, Num_eigen, p)

c      Store coefficients in a 2-D array
      call Mk_Coe(probe, Num_eigen, B, Coef)

c      Store synthetic profile in a 2-D array
      call Mk_Syn(probe, e_n_dep, p, pp)

10    continue

89    Num_probes = probe - 1
      return
      end
c ****
c subroutine EigMea(lun, N, E, e_n_dep, e_depth, fbar)

c Purpose: To load the Mean vector, and the Eigenvectors

      parameter (Maxz = 50)
      double precision E(Maxz, Maxz), fbar(Maxz)
      integer lun, N, e_n_dep
      real e_depth(Maxz)

c      Load the Mean vector, fbar
c      End of mean profile flag set at 9999.00

      do 10 k = 1, Maxz
        read(lun, *) e_depth(k), fbar(k)
        if (e_depth(k) .gt. 9000. .and. fbar(k) .gt. 9000.) goto 20
10    continue
20    e_n_dep = k - 1

c      Load the eigenvectors, E(eigenmode, depth)
c      End of each eigenvector flag set at 9999.00

```

```

do 30 j = 1, Maxz
  do 40 k = 1, Maxz
    read(lun, *, end = 50) e_depth(k), E(j, k)
    if (e_depth(k) .gt. 9000. .and. E(j, k) .gt. 9000.) goto 31
40  continue
31  continue
30  continue

c Note that there are (j - 1) eigenvectors available
50  N = j - 1

      return
      end
c ****
subroutine Ld_Eig(N, num_eigen)

c Purpose: The user decides how many modes will be used to build
c           the coefficients (and eventually the synthetic profiles).

parameter (Maxz = 50)
integer N, num_eigen

10  format(1x, A, I3, 1x, A, A)
20  write(*, 10) 'There are', N, 'eigenvalues in the ',
*                   'input eigenvector file. You can use '
*                   'a maximum of', N, 'eigenvalues in the ',
*                   'construction of the synthetic profile.'
*                   'How many do you want to use? '
read(*,*) num_eigen

      return
      end
c ****
subroutine Ld_Pro(lun, f, f_depth, f_n_dep, eof)

c Purpose: To load a single profile

parameter (Maxz = 50, Maxpts = 6000)
double precision f(Maxpts), vals(2, Maxpts)
integer lun, f_n_dep
logical eof
real f_depth(Maxpts)

c Load a profile, f
c End of profile flag set at 9999.00

```

```

do 10 k = 1, Maxpts
    read (lun,*, end = 40) vals(1, k), vals(2, k)
    if (vals(1, k) .gt. 9000. .and.
*       vals(2, k) .gt. 9000.) goto 20
        f_depth(k) = vals(1, k)
        f(k) = vals(2, k)
10    continue

20    f_n_dep = k - 1
      goto 30

40    eof = .true.
30    continue

      return
      end
c ****
subroutine Miss(f_depth, f_n_dep, e_depth, e_n_dep, flag)

```

c Purpose: To find the missing points in a profile

```

parameter (Maxz = 50, Maxpts = 6000)
integer e_n_dep, f_n_dep
logical flag(Maxz)
real e_depth(Maxz), f_depth(Maxpts)

do 10 k = 1, e_n_dep
    flag(k) = .false.
    do 20 nn = 1, f_n_dep
        if (f_depth(nn) .eq. e_depth(k)) then
            flag(k) = .true.
            goto 10
        endif
20    continue
10    continue

      return
      end
c ****

```

subroutine A_Matx(e_n_dep, E, Num_eigen, flag, A)

c Purpose: To load the "A" Matrix in $A * x = B$

```

parameter (Maxz = 50)
double precision E(Maxz, Maxz), A(Maxz, Maxz)
integer e_n_dep, Num_eigen

```

```

logical flag(Maxz)
real sum

c Create the A matrix using only the eigenvectors needed.
do 10 i = 1, Num_eigen
    do 20 j = 1, Num_eigen
        sum = 0.0
        do 30 k = 1, e_n_dep
            if (flag(k)) then
                sum = sum + E(i, k) * E(j, k)
            endif
        30     continue
        A(i, j) = sum
    20     continue
    10     continue

    return
end
c *****
subroutine Det(N, A, prod)

c Purpose: Check the determinant of the (inv)A matrix for
c           possible singularity

parameter (Maxz = 50)
double precision A(Maxz, Maxz), prod
integer N

prod = 1.0
do 10 i = 1, N
    prod = prod * A(i, i)
10    continue

4    format(1X, A, 1X, E8.2)
write(*,4) 'Determinant of the Eigenvector LUDCMP matrix =',
*          prod

    return
end
c *****
subroutine Ld_fpr(f, f_depth, f_n_dep, e_depth, e_n_dep,
*                  fbar, f_prime)

c Purpose: At each depth z, calculate f'(z) = f(z) - fbar(z)

parameter (Maxz = 50, Maxpts = 6000)

```

```

double precision fbar(Maxz), f_prime(Maxpts), f(Maxpts)
integer f_n_dep, e_n_dep
real e_depth(Maxz), f_depth(Maxpts)

do 10 k = 1, e_n_dep
  do 20 nn = 1, f_n_dep
    if (f_depth(nn) .eq. e_depth(k)) then
      f_prime(nn) = f(nn) - fbar(k)
    endif
 20  continue
10   continue

  return
end
c ****
subroutine Ld_B(E, e_depth, e_n_dep, f_depth, f_n_dep,
*                 f_prime, Num_eigen, B)

c Purpose: Load the vector "B" in A * x = B

parameter (Maxz = 50, Maxpts = 6000)
double precision B(Maxz), f_prime(Maxpts), E(Maxz, Maxz), sum
integer Num_eigen, f_n_dep, e_n_dep
real e_depth(Maxz), f_depth(Maxpts)

do 10 i = 1, Num_eigen
  sum = 0.0
  do 20 k = 1, e_n_dep
    do 30 nn = 1, f_n_dep
      if (f_depth(nn) .eq. e_depth(k)) then
        sum = sum + f_prime(nn) * E(i, k)
      endif
    30  continue
 20  continue
  B(i) = sum
10   continue

  return
end

c ****
subroutine Syn(C, E, fbar, e_n_dep, Num_eigen, p)

```

c Purpose: At each depth z, calculate the synthetic profile, p
 c p(z) = Mean(z) + sum (Coefficient * Evec)
 c Note: This is for a single profile

```

parameter (Maxz = 50)
double precision C(Maxz), fbar(Maxz), E(Maxz, Maxz), p(Maxz), sum
integer Num_eigen, e_n_dep

c Synthetic(z) = Mean(z) + sum (Coeff * Eigenvector)
do 20 k = 1, e_n_dep
    sum = 0.0
    do 10 j = 1, Num_eigen
        sum = sum + C(j) * E(j, k)
10    continue
    p(k) = fbar(k) + sum
20    continue

    return
end
c ****
subroutine Mk_Coe(probe, Num_eigen, B, Coef)

c Purpose: To store the calculated coefficients in a 2-D matrix

parameter (Maxz = 50, MaxProfiles = 100)
double precision B(Maxz), Coef(MaxProfiles, Maxz)
integer probe, Num_eigen

do 10 k = 1, num_eigen
    Coef(probe, k) = B(k)
10    continue

    return
end
c ****
subroutine Mk_Syn(probe, e_n_dep, p, pp)

c Purpose: To store the calculated synthetic profiles in a 2-D matrix

parameter (Maxz = 50, MaxProfiles = 100)
double precision p(Maxz), pp(MaxProfiles, Maxz)
integer probe, e_n_dep

do 10 k = 1, e_n_dep
    pp(probe, k) = p(k)
10    continue

    return
end
c ****

```

```
subroutine LUDCMP(N, A, indx)
```

```
c Purpose: To solve a Matrix A into L U Decompostion form  
c ... from NUMERICAL RECIPES
```

```
parameter (Maxz = 50)  
parameter (nmax = 100, TINY = 1.0E-20)  
double precision A(Maxz, Maxz), VV(nmax), aamax, sum, dum  
integer N, indx(Maxz), D, imax  
  
do 26 i = 1, N  
    indx(i) = 0  
26 continue  
  
D=1  
do 12 i=1,N  
    aamax=0.  
    do 11 j=1,N  
        if (dabs(A(i, j)) .gt. aamax) aamax = dabs(A(i, j))  
11    continue  
    if (aamax .eq. 0.) pause 'Singular matrix.'  
    VV(i) = 1. / aamax  
12    continue  
do 19 j = 1, N  
    if (j .gt. 1) then  
        do 14 i = 1, j - 1  
            sum = A(i, j)  
            if (i .gt. 1) then  
                do 13 k = 1, i - 1  
                    sum = sum - A(i, k) * A(k, j)  
13            continue  
            A(i, j) = sum  
        endif  
14    continue  
    endif  
    aamax = 0.  
    do 16 i = j, N  
        sum = A(i, j)  
        if (j .gt. 1) then  
            do 15 k = 1, j - 1  
                sum = sum - A(i, k) * A(k, j)  
15            continue  
            A(i, j) = sum  
        endif  
        dum = VV(i) * dabs(sum)  
        if (dum .ge. aamax) then
```

```

imax = i
aamax = dum
endif
16 continue
if (j .ne. imax) then
do 17 k = 1, N
dum = A(imax, k)
A(imax, k) = A(j, k)
A(j, k) = dum
17 continue
D=-D
VV(imax) = VV(j)
endif
indx(j) = imax
if (j .ne. N) then
if (A(j, j) .eq. 0.) A(j, j) = TINY
dum = 1. / A(j, j)
do 18 i = j + 1, N
A(i, j) = A(i, j) * dum
18 continue
endif
19 continue
if (A(N, N) .eq. 0.) A(N, N) = TINY

return
end
c ****
subroutine LUBKSB(A, N, indx, B)

```

c Purpose: To solve the system $A * x = B$ for x by back substitution
c ... from NUMERICAL RECIPES

```

parameter (Maxz = 50)
double precision A(Maxz, Maxz), B(Maxz), sum
integer indx(Maxz), ll, N

ii = 0
do 12 i = 1, N
ll = indx(i)
sum = B(ll)
B(ll) = B(i)
if (ii .ne. 0) then
do 11 j = ii, i - 1
sum = sum - A(i, j) * B(j)
11 continue
elseif (sum .ne. 0.) then

```

```

    ii = i
    endif
    B(i) = sum
12  continue
do 14 i = N, 1, -1
    sum = B(i)
    if (i .lt. N) then
        do 13 j = i + 1, N
            sum = sum - A(i, j) * B(j)
13  continue
    endif
    B(i) = sum / A(i, i)
14  continue

return
end
c ****
c subroutine Message()

```

c Purpose: To flash a warning message if the determinant of the
 c LUDCMP matrix is too small

```

write(*,*) ''
write(*,*) 'WARNING. In trying to calculate coefficients, it ',
*           'was found that the'
write(*,*) 'eigenvector matrix was singular; the coefficients ',
*           'may be erroneous.'
write(*,*) 'It is suggested that you re-run this program and ',
*           'choose a smaller number'
write(*,*) 'of eigenvalues to construct the coefficients.'
write(*,*) 'Processing continues...'
write(*,*) ''

```

```

return
end

```

Distribution List

Commander
Code 64H
RDT & E Division
NCCOSC
San Diego, CA 92152-5000

Dr. Richard Doolittle
Office of Naval Technology, Code 230
800 N. Quincy Street
Arlington, VA 22217-5000

Dr. Eric Hartwig
Code 7000
Naval Research Laboratory
Washington, DC 20375-5000

Robert Hearn
Code 7104
RTD & E Division
NCCOSC
San Diego, CA 92152-5000

Dr. Richard Heitmeyer
Code 5120
Naval Research Laboratory
Washington, DC 20375-5000

Dr. Bruce Howe
Applied Physics Laboratory
U of Washington
1013 NE 40th Street
Seattle, WA 98107

Dr. William Hodgkiss
Marine Physical Laboratory
Scripps Institution of Oceanography
San Diego, CA 92152

Dr. Robert Knox
Scripps Institution of Oceanography,
A-013
University of California at San Diego
La Jolla, CA 92093

Dr. Jim Mercer
Applied Physics Laboratory
U of Washington
1013 NE 40th Street
Seattle, WA 98107

Dr. Peter Worcester
Scripps Institution of Oceanography,
A-013
University of California at San Diego
La Jolla, CA 92093

NRL Code 125L (10)
Code 125P
Code 110
Code 200
Code 253 M. Fagot
Code 300
Code 322 M. Carnes
Code 330 A.W. Green
Code 331 R. Hollman
Code 331 J. Boyd (20)
Code 331 W. J. Teague
Code 331 D. Johnson

NAVOCEANO Code OPTR (attn: T.
Bennett)

Commanding Officer
Attn: Code TD Library
Naval Oceanographic Office
SSC, MS 39522-5001

Commander
Naval Oceanography Command
SSC, MS 39529-5000

Commanding Officer
Fleet Numerical Oceanography Center
Monterey, CA 93943-5005

Superintendent
Naval Postgraduate School
Monterey, CA 93943

Director
Woods Hole Oceanographic Institution
PO Box 32
Woods Hole, MA 02543

University of California
Scripps Institute of Oceanography
PO Box 6049
San Diego, CA 92106

Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Road
Laurel, MD 20707

University of Washington
Applied Physics Laboratory
1013 NE 40th St.
Seattle, WA 98105

Naval Research Laboratory
Attn: Library
Washington, DC 20375-5000

REPORT DOCUMENTATION PAGE

*Form Approved
OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. Agency Use Only (Leave blank).	2. Report Date.	3. Report Type and Dates Covered.	
	August 1992	Final	
4. Title and Subtitle.		5. Funding Numbers.	
Estimation of EOF Expansion Coefficients from Incomplete Data		Program Element No. 0603254N/ Project No. 0602314N 1401,1404,3503, 3504 Task No. JOD Accession No. DN258033 Work Unit No. 13312A	
6. Author(s).			
J.D. Boyd, E.P. Kennelly*, and P. Pistek			
7. Performing Organization Name(s) and Address(es).		8. Performing Organization Report Number.	
Naval Oceanographic and Atmospheric Research Laboratory Ocean Science Directorate Stennis Space Center, MS 39529-5004		NOARL Technical Note 291	
9. Sponsoring/Monitoring Agency Name(s) and Address(es).		10. Sponsoring/Monitoring Agency Report Number.	
Office of Naval Technology 800 N. Quincy Street Arlington, VA 22217-5000		NOARL Technical Note 291	
11. Supplementary Notes.			
*Neptune Sciences, Inc. Slidell, LA 70458			
12a. Distribution/Availability Statement.		12b. Distribution Code.	
Approved for public release; distribution is unlimited.			
13. Abstract (Maximum 200 words).			
A least-squares minimization technique for estimating principal components (empirical orthogonal function expansion coefficients) from incomplete ocean profile data is developed, and its use for interpolating or extrapolating these profiles is described.			
14. Subject Terms.		15. Number of Pages.	
Acoustic Arrays, Oceanographic Data, Underwater Sound Signals		41	
		16. Price Code.	
17. Security Classification of Report.	18. Security Classification of This Page.	19. Security Classification of Abstract.	20. Limitation of Abstract.
Unclassified	Unclassified	Unclassified	SAR